

I claim:

1. An approximate message authentication code generated by a cryptography device and comprising a probabilistic checksum generated using as input a message and a shared key and which provides absolute authentication for an origin of the message and approximate integrity for the content of the message.
2. The approximate message authentication code of claim 1, wherein the probabilistic checksum also uses an initial value as input.
3. The approximate message authentication code of claim 1, wherein data in the message are permuted.
4. The approximate message authentication code of claim 3, wherein all of the data in the message are permuted.
5. The approximate message authentication code of claim 3, wherein less than all of the data in the message are permuted.
6. The approximate message authentication code of claim 5, wherein a pseudo-random function is used to permute the data.
7. The approximate message authentication code of claim 5, wherein a random sample of data in the message are permuted.
8. The approximate message authentication code of claim 5, wherein at least one of statistical data and averages of data in the message are permuted.
9. The approximate message authentication code of claim 3, wherein the permuted data are masked into an unbiased, independent, identically distributed set of bits.

5

10

15

20

25

30

0
45
50
55
60
65
70
75
80
85
90
95

10. The approximate message authentication code of claim 3, wherein the permuted data are masked.

5 11. The approximate message authentication code of claim 10, wherein the data are masked using stream encryption.

10 12. The approximate message authentication code of claim 10, wherein:

 a. the masked data are taken in groups of rows, each row having columns of $|A|$ bits, where A is an integer, and a majority bit value is determined for each column in each group of rows to generate a new group of rows of majority bits; and

 b. a majority bit value is determined for each of $|A|$ columns of the new group of rows of majority bits.

15 13. The approximate message authentication code of claim 12, wherein the groups of rows are all groups of T rows, where T is an integer.

20 14. The approximate message authentication code of claim 12, wherein all of the groups of rows do not have the same number of rows.

25 15. The approximate message authentication code of claim 1, wherein the message is a received message and a value of the approximate message authentication code is compared with a value of second approximate message authentication code received with the message.

30 16. The approximate message authentication code of claim 15, wherein the received message is determined to have acceptable integrity if the approximate message authentication code is one of (1) a same value as and (2) within an acceptable distance of, the second approximate message authentication code.

35 17. A method performed by a cryptography device for generating an approximate message

authentication code, comprising the steps of:

- a. receiving a message containing data and arranging the data into a table having $|A|$ columns and T^2 rows, where A and T are integers;
- b. permuting at least some of the arranged data;
- c. masking the permuted data;
- d. copying the permuted and masked data into T S-arrays, each S-array having $|A|$ columns, and determining a majority bit value of each of the $|A|$ columns for each of the T S-arrays;
- e. using the determined majority bits to create a T -array having $|A|$ columns and T rows; and
- f. determining the majority value of each of the $|A|$ columns in the T array.

- 10 18. The method of claim 17, further comprising the step of generating a pseudo-random bit string before the step of permuting.
- 15 19. The method of claim 18, wherein the step of generating the pseudo-random bit string (PRBS) further comprises using a shared key and pseudo-random number generator to generate the PRBS.
- 20 20. The method of claim 19, wherein the step of generating the PRBS further comprises using an initial value to generate the PRBS.
- 25 21. The method of claim 17, further comprising the step of selecting T to be an odd integer.
- 22. The method of claim 17, further comprising the step of selecting a length of $|A|$.
- 23. The method of claim 17, wherein the step of permuting further comprises permuting the data by row.
- 30 24. The method of claim 23, further comprising the step of for each permuted row, permuting

data within each row.

5

25. The method of claim 24, wherein the step of permuting data within each row further comprises the step of circularly shifting each permuted row a pseudo-random number of places.
26. The method of claim 17, wherein the step of permuting further comprises permuting the data by bit.
- 10 27. The method of claim 17, wherein the step of permuting further comprises selecting an unpredictable permutation.
28. The method of claim 27, wherein the step of selecting an unpredictable permutation further comprises using one of a block cipher and a conventional message authentication code.
- 15 29. The method of claim 17, wherein the step of permuting comprises permuting all of the data in the message
30. The method of claim 17, wherein the step of permuting comprises permuting less than all of the data in the message.
- 20 31. The method of claim 30, wherein the step of permuting further comprises using a pseudo-random function to select the data for permuting.
- 25 32. The method of claim 30, wherein the step of permuting further comprises permuting a random sample of data in the message.
33. The method of claim 30, wherein the step of permuting further comprises permuting at least one of statistical data and averages of data in the message.

0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30

34. The method of claim 17, wherein the step of masking further comprises the step of stream encrypting the permuted data.

35. The method of claim 17, wherein the step of masking further comprises bitwise exclusive-ORing the permuted data and at least a portion of the pseudo-random bit stream.

36. The method of claim 17, wherein the step of masking comprises generating an unbiased, independent, identically distributed set of 1s and 0s.

37. The method of claim 17, wherein the step of copying the permuted and masked data into S-arrays further comprises selecting each S-array to have T rows.

38. The method of claim 37, wherein the step of copying the permuted and masked data into S-arrays further comprises adding a row of pseudo-random bits to the S-array if T is an even number.

39. The method of claim 17, wherein the step of copying the permuted and masked data into S-arrays further comprises not selecting each S-array to have the same number of rows.

40. A device for generating an approximate message authentication code, comprising:

- a pseudo-random bit string generator module configured to receive as input a secret key and to output a string of pseudo-random bits;
- an arrangement module configured to receive a message containing data and arrange the data into a table having $|A|$ columns and T^2 rows, where A and T are integers;
- a permuting module responsive to the arranged data and at least a portion of the string of pseudo-random bits and configured to permute arranged data;
- a masking module responsive to the permuting module and at least a portion of the string of pseudo-random bits and configured to mask the randomized data; and
- a majority module responsive to the masking module and configured to:

5

- i. copy the masked data into T S-arrays, each array having $|A|$ columns, and to determine the majority bit value of each of the $|A|$ columns for each of the S-arrays;
- ii. use the determined majority bits to create a T -array having $|A|$ columns and T rows; and
- iii. determine the majority value of each of the $|A|$ columns in the T array.

10

41. The device of claim 40, wherein the pseudo-random bit string generator module further comprises a pseudo-random number generator.

15

42. The device of claim 41, wherein the pseudo-random number generator is a cryptographically secure random number generator.

43. The device of claim 40, wherein the pseudo-random bit string generator module is further configured to receive as input an initial value.

20

44. The device of claim 40, wherein T is an odd integer.

25

45. The device of claim 40, wherein $|A|$ is selected to have a predetermined length, the selected length depending on a sensitivity to bit changes.

30

46. The device of claim 40, wherein the permuting module is configured to permute the data by row.

47. The device of claim 46, wherein the permuting module is further configured, for each permuted row, to permute data within each row.

48. The device of claim 47, wherein the permuting module is further configured to circularly shift each permuted row a pseudo-random number of places.

30
25
20
15
10
5

49. The device of claim 40, wherein the permuting module is configured to permute the data by one of bit, byte, and data word.
50. The device of claim 40, wherein the permuting module is further configured to use a collision-free, unpredictable permutation to permute the data.
51. The device of claim 40, wherein the permuting module is configured to permute all of the data in the message
52. The device of claim 40, wherein the permuting module is configured to permute less than all of the data in the message.
53. The device of claim 52, wherein permuting module uses a pseudo-random function to select the data for permuting.
54. The device of claim 52, wherein the permuting module is configured to permute a random sample of data in the message.
55. The device of claim 52, wherein the permuting module is configured to permute at least one of statistical data and averages of data in the message.
56. The device of claim 40, wherein the masking module further comprises an exclusive-OR circuit responsive to the permuting module and at least a portion of the string of pseudo-random bits.
57. The device of claim 40, wherein the majority module is configured to copy the masked data into S-arrays each having the same number of rows.
58. The device of claim 40, wherein the majority module is configured to copy the masked data into S-arrays not all having the same number of rows.

5

10

15

20

25

30

35

40

45

50

55

60

65

70

75

80

85

90

95

100

105

110

115

120

125

130

135

140

145

150

155

160

165

170

175

180

185

190

195

200

205

210

215

220

225

230

235

240

245

250

255

260

265

270

275

280

285

290

295

300

305

310

315

320

325

330

335

340

345

350

355

360

365

370

375

380

385

390

395

400

405

410

415

420

425

430

435

440

445

450

455

460

465

470

475

480

485

490

495

500

505

510

515

520

525

530

535

540

545

550

555

560

565

570

575

580

585

590

595

600

605

610

615

620

625

630

635

640

645

650

655

660

665

670

675

680

685

690

695

700

705

710

715

720

725

730

735

740

745

750

755

760

765

770

775

780

785

790

795

800

805

810

815

820

825

830

835

840

845

850

855

860

865

870

875

880

885

890

895

900

905

910

915

920

925

930

935

940

945

950

955

960

965

970

975

980

985

990

995

1000

1005

1010

1015

1020

1025

1030

1035

1040

1045

1050

1055

1060

1065

1070

1075

1080

1085

1090

1095

1100

1105

1110

1115

1120

1125

1130

1135

1140

1145

1150

1155

1160

1165

1170

1175

1180

1185

1190

1195

1200

1205

1210

1215

1220

1225

1230

1235

1240

1245

1250

1255

1260

1265

1270

1275

1280

1285

1290

1295

1300

1305

1310

1315

1320

1325

1330

1335

1340

1345

1350

1355

1360

1365

1370

1375

1380

1385

1390

1395

1400

1405

1410

1415

1420

1425

1430

1435

1440

1445

1450

1455

1460

1465

1470

1475

1480

1485

1490

1495

1500

1505

1510

1515

1520

1525

1530

1535

1540

1545

1550

1555

1560

1565

1570

1575

1580

1585

1590

1595

1600

1605

1610

1615

1620

1625

1630

1635

1640

1645

1650

1655

1660

1665

1670

1675

1680

1685

1690

1695

1700

1705

1710

1715

1720

1725

1730

1735

1740

1745

1750

1755

1760

1765

1770

1775

1780

1785

1790

1795

1800

1805

1810

1815

1820

1825

1830

1835

1840

1845

1850

1855

1860

1865

1870

1875

1880

1885

1890

1895

1900

1905

1910

1915

1920

1925

1930

1935

1940

1945

1950

1955

1960

1965

1970

1975

1980

1985

1990

1995

2000

2005

2010

2015

2020

2025

2030

2035

2040

2045

2050

2055

2060

2065

2070

2075

2080

2085

2090

2095

2100

2105

2110

2115

2120

2125

2130

2135

2140

2145

2150

2155

2160

2165

2170

2175

2180

2185

2190

2195

2200

2205

2210

2215

2220

2225

2230

2235

2240

2245

2250

2255

2260

2265

2270

2275

2280

2285

2290

2295

2300

2305

2310

2315

2320

2325

2330

2335

2340

2345

2350

2355

2360

2365

2370

2375

2380

2385

2390

2395

2400

2405

2410

2415

2420

2425

2430

2435

2440

2445

2450

2455

2460

2465

2470

2475

2480

2485

2490

2495

2500

2505

2510

2515

2520

2525

2530

2535

2540

2545

2550

2555

2560

2565

2570

2575

2580

2585

2590

2595

2600

2605

2610

2615

2620

2625

2630

2635

2640

2645

2650

2655

2660

2665

2670

2675

2680

2685

2690

2695

2700

2705

2710

2715

2720

2725

2730

2735

2740

2745

2750

2755

2760

2765

2770

2775

2780

2785

2790

2795

2800

2805

2810

2815

2820

2825

2830

2835

2840

2845

2850

2855

2860

2865

2870

2875

2880

2885

2890

2895

2900

2905

2910

2915

2920

2925

2930

2935

2940

2945

2950

2955

2960

2965

2970

2975

2980

2985

2990

2995

3000

3005

3010

3015

3020

3025

3030

3035

3040

3045

3050

3055

3060

3065

3070

3075

3080

3085

3090

3095

3100

3105

3110

3115

3120

3125

3130

3135

3140

3145

3150

3155

3160

3165

3170

3175

3180

3185

3190

3195

3200

3205

3210

3215

3220

3225

3230

3235

3240

3245

3250

3255

3260

3265

3270

3275

3280

3285

3290

3295

3300

3305

3310

3315

3320

3325

3330

3335

3340

3345

3350

3355

3360

3365

3370

3375

3380

3385

3390

3395

3400

3405

3410

3415

3420

3425

3430

3435

3440

3445

3450

3455

3460

3465

3470

3475

3480

3485

3490

3495

3500

3505

3510

3515

3520

3525

3530

3535

3540

3545

3550

3555

3560

3565

3570

3575

3580

3585

3590

3595

3600

3605

3610

3615

3620

3625

3630

3635

3640

3645

3650

3655

3660

3665

3670

3675

3680

3685

3690

3695

3700

3705

3710

3715

3720

3725

3730

3735

3740

3745

3750

3755

3760

3765

3770

3775

3780

3785

3790

3795

3800

3805

3810

3815

3820

3825

3830

3835

3840

3845

3850

3855

3860

3865

3870

3875

3880

3885

3890

3895

3900

3905

3910

3915

3920

3925

3930

3935

3940

3945

3950

3955

3960

3965

3970

3975

3980

3985

3990

3995

4000

4005

4010

4015

4020

4025

4030

4035

4040

4045

4050

4055

4060

4065

4070

4075

4080

4085

4090

4095

4100

4105

4110

4115

4120

4125

4130

4135

4140

4145

4150

4155

4160

4165

4170

4175

4180

4185

4190

4195

4200

4205

4210

4215

4220

4225

4230

4235

4240

4245

4250

4255

4260

4265

4270

4275

4280

4285

4290

4295

4300

4305

4310

4315

4320

4325

4330

4335

4340

4345

4350

4355

4360

4365

4370

4375

4380

4385

4390

4395

4400

4405

4410

4415

4420

4425

4430

4435

4440

4445

4450

4455

4460

4465

4470

4475

4480

4485

4490

4495

4500

4505

4510

4515

4520

4525

4530

4535

4540

4545

4550

4555

4560

4565

4570

4575

4580

4585

4590

4595

4600

4605

4610

4615

4620

4625

4630

4635

4640

4645

4650

4655

4660

4665

4670

4675

4680

4685

4690

4695

4700

4705

4710

4715

4720

4725

4730

4735

4740

4745

4750

4755

4760

4765

4770

4775

4780

4785

4790

4795

4800

4805

4810

4815

4820

4825

4830

4835

4840

4845

4850

4855

4860

4865

4870

4875

4880

4885

4890

4895

4900

4905

4910

4915

4920

4925

4930

4935

4940

4945

4950

4955

4960

4965

4970

4975

4980

4985

4990

4995

5000

5005

5010

5015

5020

5025

5030

5035

5040

5045

5050

5055

5060

5065

5070

5075

5080

5085

5090

5095

5100

5105

5110

5115

5120

5125

5130

5135

5140

5145

5150

5155

5160

5165

5170

5175

5180

5185

5190

5195

5200

5205

5210

5215

5220

5225

5230

5235

5240

5245

5250

5255

5260

5265

5270

5275

5280

5285

5290

5295

5300

5305

5310

5315

5320

5325

5330

5335

5340

5345

5350

5355

5360

5365

5370

5375

5380

5385

5390

5395

5400

5405

5410

5415

5420

5425

5430

5435

5440

5445

5450

5455

5460

5465

5470

5475

5480

5485

5490

5495

5500

5505

5510

5515

5520

5525

5530

5535

5540

5545

5550

5555

5560

5565

5570

5575

5580

5585

5590

5595

5600

5605

5610

5615

5620

5625

5630

5635

5640

5645

5650

5655

5660

5665

5670

5675

5680

5685

5690

5695

5700

5705

5710

5715

5720

5725

5730

5735

5740

5745

5750

5755

5760

5765

5770

5775

5780

5785

5790

5795

5800

5805

5810

5815

5820

5825

5830

5835

5840

5845

5850

5855

5860

5865

5870

5875

5880

5885

5890

5895

5900

5905

5910

5915

5920

5925

5930

5935

5940

5945

5950

5955

5960

5965

5970

5975

5980

5985

5990

5995

6000

6005

6010

6015

6020

6025

6030

6035

6040

6045

6050

6055

6060

6065

6070

6075

6080

6085

6090

6095

6100

6105

6110

6115

6120

6125

6130

6135

6140

6145

6150

6155

6160

6165

6170

6175

6180

6185

6190

6195

6200

6205

6210

6215

6220

6225

6230

6235

6240

6245

6250

6255

6260

6265

6270

6275

6280

6285

6290

6295

6300

6305

6310

6315

6320

6325

6330

6335

6340

6345

6350

6355

6360

6365

6370

6375

6380

6385

6390

6395

6400

6405

6410

6415

6420

6425

6430

64

15
16
17
18
19
20

5

- array in the second AMAC;
- c. using the distribution of Hamming weights, determining a probability that d differences in a column of an array will cause a change in a majority bit value in that column;
- d. using the likelihood that bit differences will occur in a column of an instance of an S-array in the second AMAC and the probability that d differences in a column of an array will cause a change in a majority bit value in that column, determining an expected number of differences between a T -array for the first AMAC and a T -array for the second AMAC;
- 10 e. estimating a distribution of the d differences across columns of the T -array for the second AMAC; and
- f. estimating for a second time the distribution of the d differences across columns of the T -array for the second AMAC.

62. The method of claim 61, wherein the step of determining a likelihood that bit differences will occur in a column of an instance of an S-array in the second AMAC further comprises using a hyper-geometric distribution.

63. The method of claim 61, wherein the step of determining a distribution of Hamming weights of columns of an instance of an S-array further comprises determining a binomial distribution.

64. The method of claim 61, wherein the step of determining a probability that d differences in a column of an array will cause a change in a majority bit value in that column further comprises by determining all cases that can possibly change the majority bit value.

25

65. The method of claim 61, wherein the step of estimating a distribution of the d differences across columns of the T -array for the second AMAC further comprises using Bose-Einstein occupancy statistics.